



TITLE:

非線形項書き換えシステムのE-非オーバーラップ性について(理論計算機科学とその周辺)

AUTHOR(S):

張, 磊; 大山口, 通夫; 伊藤, 英則

CITATION:

張, 磊 ...[et al]. 非線形項書き換えシステムのE-非オーバーラップ性について(理論計算機科学とその周辺). 数理解析研究所講究録 1992, 790: 263-268

ISSUE DATE:

1992-06

URL:

<http://hdl.handle.net/2433/82641>

RIGHT:

非線形項書き換えシステムのE-非オーバーラップ性について

張 磊 大山口 通夫* 伊藤 英則

名古屋工業大学

三重大学*

1. はじめに

これまで、項書き換えシステム (TRS) に関する研究は主に、線形或いは停止性を有する場合について行われてきたが、最近では、非線形かつ非停止の場合についても、新しい結果が幾つか報告されるようになってきた[3, 5, 6]. 文献[3]では、E-非オーバーラップ性の概念が提案され、E-非オーバーラップ性を満たすTRSにおいては、各項の正規形は唯一であることが示された。

本研究では、このE-非オーバーラップ性について考察し、ある制限されたTRSのクラスでは、“ ω -非オーバーラップ性質を持つ”という定理を証明することによって、E-非オーバーラップ性を保証する、一つの判定可能な十分条件を与える。この結果の系として、非線形かつ非停止のTRSが合流性(Church-Rosser)を満たすための一つの判定可能な十分条件を示す。

2. 諸定義

TRSに関する基本的な定義については文献[1-4]を参照されたい。非線形TRSとは、書換え規則の左辺に同じ変数の出現が二度以上に許されるTRSである。非線形TRSが制限的とは、右線形であり、かつ、規則の左辺に同じ変数がたかだか二度出現するときである。ここで、TRSが右線形とは、規則の右辺に出現する変数はたかだか一度出現するときである。

項書き換えシステムRの規則のペア $\alpha_i \rightarrow \beta_i$ と $\alpha_j \rightarrow \beta_j$ がE-非オーバーラップであるとは文脈C[], 非変数の項M及び代入 θ が存在して、 $\alpha_i \equiv C[M]$, $\theta(\alpha_j) =_M \theta(M)$ が成立するときである。但し、項のトップを書き換えることはない。

TRS Rは $\langle E, n \rangle$ -E-非オーバーラップであるというのは、規則 $\alpha_i \rightarrow \beta_i$, $\alpha_j \rightarrow \beta_j \in R$, 文脈C[], 非変数の項Mと代入 θ が存在して、次の条件が成立するときである。(但し、 $i \neq j$, 又は $C \neq []$) $\alpha_i \equiv C[M]$, かつ、 $\theta(\alpha_j) =_M \theta(M)$ である。但し、項のトップ書き換えることはなく、書き換えの回数がn以下である。このようなペア $\langle \alpha_i, M \rangle$ を $\langle E, n \rangle$ -E-非オーバーラップ \wedge と呼ぶ。

項書き換えシステムRがE-非オーバーラップシステムであるとは、自明な場合(つまり: $i = j \wedge C[] = []$)を除いて、任意の二つの規則がE-非オーバーラップでないときである。

TRS Rの規則の組が ω -E-非オーバーラップ性質を満たすとは無限項の代入を許して、 ω -E-非オーバーラップするときである。 ω -E-非オーバーラップシステムとはTRS Rの任意の二つの規則はその[閉]のケースを除いて、 ω -E-非オーバーラップではないシステムである。

本稿では、制限的な非線形TRSがE-非オーバーラップ性を満たすための十分条件を与える。なお、TRSがE-非オーバーラップ性を満たすかどうか判定する問題は、一般に非可解な問題であることが知られている([3])。

3. 主定理

[定理1] TRS Rは制限的であるとするとき、そのとき、Rは ω -E-非オーバーラップ性質をもつならば、E-非オーバーラップ性質を持つ。

この証明は背理法による。即ち、もし、ある $\langle E, n \rangle$ -E-非オーバーラップ \wedge が存在するならば、 ω -E-非オーバーラップ性質を満たさないことを示すことによって、矛盾を導く。

矛盾を導くために、次節で導入する ω -Unifyアルゴリズムを必要とする。そして、この ω -Unifyアルゴリズムが次の二つの条件(1)と(2)を満たすことを示す。

- (1). ω -Unifyアルゴリズムは ω -単一化アルゴリズムである。
- (2). ある $\langle E, n \rangle$ -E-非オーバーラップ \wedge に対して、 ω -Unifyアルゴリズムは成功する(即ち、 ω -単一化に成功する)。

以上の(1)と(2)より、TRS Rが ω -E-非オーバーラップ性質をもつならば、 $\langle E, n \rangle$ -E-非オーバーラップ \wedge の存在が否定される。ゆえに、定理1が成立する。

4. ω -Unifyアルゴリズム

このアルゴリズムに使われる定義と記法を簡単に述べる。

[定義] 変数と項の組からなる全集合を $P(V, T)$ とする。但し、変数と項の集合をそれぞれVとTとする。

[記法] 集合 $B \subseteq P(V, T)$ に関するインサート操作は以下のように定義される。

$\text{Insert}(B, \{(x_1, P_1); \dots; (x_n, P_n)\}) \equiv \{(y, \sigma(Q)) \mid (y, Q) \in B \cup \{(x_1, P_1); \dots; (x_n, P_n)\}\}$

但し、写像 $\sigma = [P_1/x_1, \dots, P_n/x_n]$, $(x_i, P_i) \in P(V, T)$

[定義] 項M, Nの変数の出現の中の極小出現の集合 $V_{M, N}$ は以下

On the E-nonoverlapping Property of Nonlinear Term Rewriting Systems

Lei Zhang, Michio Oyamaguchi* and Hidenori Itoh
Nagoya Institute of Technology

* Mie University

のように定義される. $V_{\text{min}}(M, N) \equiv \min(0, (M) \cup 0, (N))$

ここで, $0, (M) = \{u \mid M/u \in V\}$ 即ち, 変数出現の集合である.

【記法】 操作 $\text{Front}(M, N)$ は項 M, N のフロンティアを取る.
フロンティアを二つに分けて, A は変数の \wedge の集合であり, S は
その以外の \wedge の集合である.

$\text{Front}(M, N) \equiv \{(M/u, N/u) \mid u \in V_{\text{min}}(M, N)\}$

$C(M, N) \equiv V \times V \cap \text{Front}(M, N) \subset A$ (A は変数の \wedge の集合である)

$S(M, N) \equiv \text{Front}(M, N) - C(M, N)$

$\equiv \{(M_j, N_j) \mid M/u_j = M_j \in V \wedge N/u_j = N_j \in T; u_j \in V_{\text{min}}(M, N)\}$

【定義】 操作 $\text{Check}(M, N)$ は項 M, N の非変数部分の整合性をチェックする. つまり, 以下のように定義される.

$M[u \leftarrow \Omega \mid u \in V_{\text{min}}(M, N)] = N[u \leftarrow \Omega \mid u \in V_{\text{min}}(M, N)]$ であれば,

$\text{Check}(M, N) \equiv \text{true}$ である.

$M[u \leftarrow \Omega \mid u \in V_{\text{min}}(M, N)] \neq N[u \leftarrow \Omega \mid u \in V_{\text{min}}(M, N)]$ であれば,

$\text{Check}(M, N) \equiv \text{failure}$ である.

【記法】 操作 $\text{Now}(x)$ は x に新しい変数名を割り当てる手続きである. ここで, 新しい変数名とはその時点でプログラムの中で使われていない変数である.

【記法】 操作 $\max(M, N)$ と $\min(M, N)$ は以下のである.

$\max(M, N) = M$ ($N \in V$ のとき)

or $= N$ ($M \in V$ のとき)

$\min(M, N) = M$ ($N \in T$ のとき)

or $= N$ ($M \in T$ のとき)

アルゴリズム $\omega\text{-Unify}(M, N)$;

begin

(I) if $\text{Check}(M, N) = \text{FALSE}$ then return(failure);

(II) $\text{Eq} := S(M, N)$;

$A := C(M, N)$;

$B := \emptyset$;

(III) while $(\exists (x, P), (x, Q) \in B)$ do

begin

$B := \{B - \{(x, P), (x, Q)\} \cup \{(x, P[u_j \leftarrow \max(M/u_j, N/u_j) \mid$

$u_j \in V_{\text{min}}(P, Q)]\}$

$B := \text{Insert}(B, \text{Front}(P, Q))$

end;

(IV) while $(\exists (x, P) \in B, (x, Q) \in \text{Eq})$ do

begin

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

$\text{Eq} := (\text{Eq} - \{(x, Q)\}) \cup S(P, Q)$;

$A := A \cup C(P, Q)$

end;

(V) if $(\exists (x, P), (x, Q) \in \text{Eq}; P \neq Q)$ then

begin

if $V(P) \cup V(Q) = \emptyset$ then return(failure);

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

$B := \text{Insert}(B, \{(x, P[u_j \leftarrow x_j \mid u_j \in V_{\text{min}}(P, Q); \text{Now}(x_j),$

$1 \leq j \leq |V_{\text{min}}(P, Q)|\})\}$;

$\text{Eq} := \{\text{Eq} - \{(x, P), (x, Q)\}\}$

$\text{ADD}(x_j, P/u_j) ; (1 \leq j \leq |V_{\text{min}}(P, Q)|)$

$A := A \cup \{(x_j, Q/u_j) \mid 1 \leq j \leq |V_{\text{min}}(P, Q)|\}$

goto (III)

end;

(VI) if $(\exists x \rightarrow_{\Lambda} y, (x, P) \in B, (y, Q) \in \text{Eq}; x \neq y)$ then

begin

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

if $V(P) \cup V(Q) = \emptyset$ then

begin

$B := \text{Insert}(B, \{y, P\})$;

$\text{Eq} := \text{Eq} - \{y, Q\}$

else begin

$B := \text{Insert}(B, \{(y, Q[u_j \leftarrow y_j \mid u_j \in V_{\text{min}}(P, Q);$

$\text{Now}(y_j), 1 \leq j \leq |V_{\text{min}}(P, Q)|\})\}$;

$\text{Eq} := \text{Eq} - \{y, Q\}$

$\text{ADD}(y_j, Q/u_j \mid 1 \leq j \leq |V_{\text{min}}(P, Q)|)$

end

goto (III)

end;

(VII) if $(\exists x \rightarrow_{\Lambda} z \rightarrow_{\Lambda} y, (x, P) \in B, (y, Q) \in B, P \neq Q)$ then

begin

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

$A := A - \{(x, z)\}$;

$B := \text{Insert}(B, \{(z, P[u_j \leftarrow z_j \mid u_j \in V_{\text{min}}(P, Q); \text{Now}(z_j),$

$1 \leq j \leq |V_{\text{min}}(P, Q)|\})\}$;

$\text{ADD}(z_j, P/u_j \mid 1 \leq j \leq |V_{\text{min}}(P, Q)|)$

if $(\exists x \rightarrow y, (x, P) \in B, (y, Q) \in B, P \neq Q)$ then

begin

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

$A := A - \{(x, y)\}$;

$\text{ADD}(\min(P/u_j, Q/u_j), \max(P/u_j, Q/u_j))$

$(1 \leq j \leq |V_{\text{min}}(P, Q)|)$

goto (III)

end;

(VIII) if $(\exists x \rightarrow_{\Lambda} y, (x, P) \in \text{Eq}, (y, Q) \in \text{Eq}, x \neq y)$ then

begin

if $V(P) \cup V(Q) = \emptyset \wedge P \neq Q$ then return(failure);

if $V(P) \cup V(Q) = \emptyset \wedge P = Q$ then

$B := \text{Insert}(B, \{(x, P), (y, Q)\})$;

$\text{Eq} := \text{Eq} - \{(x, P), (y, Q)\}$;

if $V(P) \cup V(Q) \neq \emptyset$ then

begin

if $\text{Check}(P, Q) = \text{FALSE}$ then return(failure);

$B := \text{Insert}(B, \{(x, P[u_j \leftarrow a_j \mid u_j \in V_{\text{min}}(P, Q); \text{Now}(a_j),$

$1 \leq j \leq |V_{\text{min}}(P, Q)|\}),$

$(y, Q[u_j \leftarrow b_j \mid u_j \in V_{\text{min}}(P, Q); \text{Now}(b_j),$

$1 \leq j \leq |V_{\text{min}}(P, Q)|\})\}$;

$\text{Eq} := (\text{Eq} - \{(x, P), (y, Q)\})$;

$\text{ADD}(a_j, P/u_j \mid 1 \leq j \leq |V_{\text{min}}(P, Q)|)$

$\text{ADD}(b_j, Q/u_j \mid 1 \leq j \leq |V_{\text{min}}(P, Q)|)$

```

end;
goto (III)
end
(IX) return(success)
end.

```

プロセスADDは以下である。

```

procedure ADD(vor, term)
begin
  if term ∈ V then
    A := A ∪ {(vor, term)};
  else
    Bq := Bq ∪ {(vor, term)};
  end.

```

ここで、このアルゴリズムは ω -単一化アルゴリズムになっている。即ち、 (M, N) が ω -単一化が可能であれば、値としてsuccessを返し、そうでなければ、failureを返すということは以下の補題1より明かである。

また、これからのすべての補題は以下の前題によるものである。

【前題】：

項書き換えシステムRが $\langle E, n \rangle$ 非オーバーラップであり、そして、 $\langle E, n+1 \rangle$ オーバーラップである。アルゴリズム ω -unifyの入力としては $\langle E, n+1 \rangle$ オーバーラップペア $\langle \theta(\alpha_i), \theta'(M) \rangle$ である。
 $(\alpha_i \rightarrow \beta_i, \alpha_j \rightarrow \beta_j \in R \quad \alpha_j / u = M \in V \quad u \in O(\alpha_j) \wedge O(M))$

【補題1】：

ω -Unifyアルゴリズムは ω -単一化アルゴリズムである。
 (1). $\langle E, n+1 \rangle$ オーバーラップペア $\langle \theta(\alpha_i), \theta(M) \rangle$ が単一化できれば、アルゴリズム ω -Unifyの出力 $E q(t) \cup A(t) \cup B(t)$ も単一化できる。

(2). (1)の逆も成立する

(3). アルゴリズム ω -Unifyの出力 $E q(t) \cup A(t) \cup B(t)$ も単一化できる。

(4). アルゴリズム ω -Unifyは停止する。

【定義】：アルゴリズム ω -Unifyのワン・ステップとは、このアルゴリズム(I), (II), (III), (IV), (V), (VI), (VII), (VIII)のどれか一回実行される事である。その中、ループになっている(III)と(IV)に対して、ループ一回廻ることをワン・ステップであるとする。

アルゴリズムの中集合 $E q, A, B$ に対して、 t ステップ後の内容をそれぞれ $E q(t), A(t), B(t)$ と書く。

【証明】：

(1)の証明：

この証明はアルゴリズム ω -Unifyのワン・ステップづつの出力が ω -単一化アルゴリズムの解になることによる。

$\langle E, n+1 \rangle$ オーバーラップペア $\langle \theta(\alpha_i), \theta(M) \rangle$ が単一化できる前題によって、このアルゴリズムの全てのチェックの所で失敗しないである。(実際、失敗しないのは補題3に証明する。)以

下、アルゴリズム ω -Unifyのステップ数 t に関する帰納法で証明する。

①. $t=2$ の時、 $\langle E, n+1 \rangle$ オーバーラップペア $\langle \theta(\alpha_i), \theta(M) \rangle$ の ω -単一化の解が集合 S と C によって、保存されている。

②. ステップ数 t の時、 ω -単一化アルゴリズムの解が $E q(t-1) \cup A(t-1) \cup B(t-1)$ に保存されている事を仮定する。

t ステップのとき考える。

(III)から(VIII)まで、同じく、入力の前フロンティア部分に対して、操作を行って、入力の前フロンティア部分が $B(t), A(t), E q(t)$ に移した。帰納の仮定によって、このアルゴリズムが停止するとき、解が $E q \cup A \cup B$ に保存される。

つまり： $E q \cup A \cup B$ は ω -単一化出来る。

(2)の証明：

(1)により、分かったのはこのアルゴリズムの操作はただフロンティア部分で行う。新しく導入されるのは変数だけである。このフロンティア部分からなる集合の ω -単一化、いつも成功するので、それらのフロンティア部分の親の方も成功しないといけない。従って、入力も単一化できる。

(3)の証明：

ここで、背理法を用いるため、アルゴリズム ω -Unifyの出力集合 $E q(t) \cup A(t) \cup B(t)$ は単一化不可能である事を仮定する。この場合、 $\exists (x, S_1), (x, S_2) \in E q \cup A \cup B$, 出現 $u \in O(S_1) \cup O(S_2)$, $root(S_1/u) \neq root(S_2/u)$ 。もし、 $S_1 \in V$ または $S_2 \in V$ なら、 (x, S_1) と (x, S_2) が単一化が出来てしまうので、 (x, S_1) と $(x, S_2) \in E q \cup B$ だけ討論すればよい。

①. $(x, S_1) \in B, (x, S_2) \in B$

こういう対が存在する限り、アルゴリズム ω -unifyの(III)が動き続ける、つまり：このケースが有り得ない。

②. $(x, S_1) \in B, (x, S_2) \in E q$

こういう対が存在するなら、アルゴリズム ω -unifyの(VI)でのcheckの所で失敗してしまい、failureを返す。しかし、補題3によって、これがあり得ない。

③. $(x, S_1) \in E q, (x, S_2) \in E q$

こういう対が存在するなら、アルゴリズム ω -unifyの(V)でのcheckの所で失敗してしまい、failureを返す。しかし、補題3によって、これがあり得ない。

従って、最終形 $E q \cup A \cup B$ が単一化可能である。

(4)の証明：

任意の時刻のサイズ $L(t) = \Sigma |\theta(x)| + |\theta(P)|$

$((x, P) \in E q \cup A)$ に関する帰納法で停止性を示そう。

①. $t \leq 2$ の時、入力 $\theta(\alpha_i)$ と $\theta(M)$ が(II)で $E q$ と A に分けられる。当然に：

$|\theta(\alpha_i)| + |\theta(M)| > |E q(0)| + |A(0)|$

入力のサイズが減った。

②. $t=1$ の時、 $L(t_{i-1}) > L(t_i)$ であり、つまり：

$|E q(t_{i-1})| + |A(t_{i-1})| > |E q(t_i)| + |A(t_i)|$ である事を仮定する。

$t = t_{i+1}$ の時、(III)から(VII)まで順番に述べる。

(III). このループの全ての操作がE qとAと関係がないから、このループが無限に続く事がないを示せばよい。

この証明はB(t)の左辺のPとQの深さD(P)とD(Q)に関する帰納法で行う：

$(x, P), (x, Q) \in B(t)$ に対して、

$\max(D(B(t))) = n \Rightarrow$ (III)が停止することを示す。

$\max(D(B(t))) = \max\{D(x), D(P) \mid (x, P) \in B\}$ かつ、xを左としてかつBの要素が二個以上である。

1). $\max(D(B(t))) = 1$ の時、つまり： $\exists (x, P), (x, Q) \in B$, PとQはそれぞれ

$f(x_1, x_2, \dots, x_k)$ と $f(y_1, y_2, \dots, y_k)$ ($x_k \in V, y_k \in V, k=1, 2, \dots$)

の形である。従って：

$\text{Front}(P, Q) = \{(x_1, y_1), \dots, (x_k, y_k)\}$

つまり、PとQの深さが一つ減った。

2). $\max(D(B(t))) < n$ の時、PとQの深さが減っていたと仮定する。

$\max(D(B(t))) = n$ の時を考えよう。

フロンティアを取る事によって、上記のようなPとQが少なくとも一つ減る事が明かである。

なお、インサート操作により、 $\text{Front}(P, Q)$ のある要素が何回にコピーされても、深さの減ったものに限られている。帰納の仮定により、こんな項の深さも減っていたので、有限のD(P)とD(Q)に対して、いつかD(P)またはD(Q)=1になり、ループが抜ける。

(VI). $L(t_i) = |\theta(x)| + |\theta(Q)|$

$L(t_{i+1}) = |\text{Front}(P, Q)| = |\{(P_k, Q_k) \mid k=1, 2, \dots\}|$

$|\theta(P)| + |\theta(Q)| > |\text{Front}(P, Q)|$

$\therefore \theta(x) = \theta(P)$ (補題3による)

$|\theta(x)| + |\theta(Q)| > |\text{Front}(P, Q)|$

つまり： $L(t_i) > L(t_{i+1})$

(V). $L(t_i) = 2|\theta(x)| + |\theta(P)| + |\theta(Q)|$

$L(t_{i+1}) = \Sigma(2|\theta(x_1)| + |\theta(N_1)| + |\theta(M_1)|)$

$|\theta(P)| + |\theta(Q)| > |\text{Front}(P, Q)|$

$\therefore \theta(x_1) = \theta(x) / u_1$

$\{\theta(N_1) + \theta(M_1)\} \subset \{\theta(P) + \theta(Q)\}$

従って： $L(t_i) > L(t_{i+1})$

(VI). $L(t_i) = |\theta(y)| + |\theta(Q)|$

1). $V(P) \cup V(Q) = \emptyset$ の時、

$E q := E q - \{(y, Q)\}$, 従って： $L(t_i) > L(t_{i+1})$

2). $V(P) \cup V(Q) \neq \emptyset$ の時、

$L(t_{i+1}) = \Sigma|\theta(y_1)| + |\theta(Q/u_1)|$

$|\theta(P)| + |\theta(Q)| > |\text{Front}(P, Q)|$

$\therefore \theta(y_1) = \theta(y) / u_1$, 従って： $L(t_i) > L(t_{i+1})$

(VII). $(x, P) \in B, (y, Q) \in B$ なので、この操作はBの内部で行うが、AとE qのサイズが増えない事を示そう。

1). $x \xrightarrow{A} z \xrightarrow{A} y$ の時：

$A := A - \{(x, z)\}$ により、

Aのサイズが $|\theta(x)| + |\theta(z)|$ を減らした。

その後の操作により、増える分は $\Sigma(|\theta(z_1)| + |\theta(P/u_1)|)$

$\therefore |\theta(x)| = |\theta(P)| > \Sigma|\theta(P/u_1)|$

Bに入れた操作により、 $|\theta(z)| = |\theta(P)| > \Sigma|\theta(z_1)|$

従って： $L(t_i) > L(t_{i+1})$

2). $x \xrightarrow{A} y$ の時：この場合はほぼ1)と同じから、省略する。

(VIII) : $\exists x \xrightarrow{A} y, (x, P) \in B, (y, Q) \in B, x \neq y$

1). $V(P) \cup V(Q) = \emptyset$ の時、

$E q := E q - \{(x, P), (y, Q)\}$

従って： $L(t_i) > L(t_{i+1})$

2). $V(P) \cup V(Q) \neq \emptyset$ の時、

$E q := E q - \{(x, P), (y, Q)\}$ でサイズが減った。

その後の操作により、増える分は

$\Sigma(|\theta(Q/u_1)| + |\theta(P/u_1)| + |\theta(a_1)| + |\theta(b_1)|)$

Bに入れた操作により、 t_{i+1} の時：

$\therefore |\theta(x)| = |\theta(P)| > \Sigma|\theta(a_1)|$

$|\theta(y)| = |\theta(Q)| > \Sigma|\theta(b_1)|$

$|\theta(P)| > \Sigma|\theta(P/u_1)|$

$|\theta(Q)| > \Sigma|\theta(Q/u_1)|$

従って： $L(t_i) > L(t_{i+1})$

以上の審議を纏めて、 t_{i+1} の時にも、 $L(t_i) > L(t_{i+1})$ が成立するから、このアルゴリズムが停止する。■

次に、3節の条件(2)を証明するために、 ω -非オーバーラップの前提条件の元で、 $\langle E, n \rangle$ -オーバーラップペア $\langle \alpha_1, M \rangle$ の具体例 $\langle \theta(\alpha_1), \theta(M) \rangle$ のうち、nが最小であるようなペアをアルゴリズム ω -Unifyに入力すると、 ω -Unifyが成功することを示すことが出来る。条件(1)よりアルゴリズム ω -unify中の整合性を検査する手続きcheckがいつも真であることを示せばよい。その証明は以下いくつかの補題によって与える。

【補題2】：

任意の部分リデックス・ペア $\langle \theta(\alpha), \theta^*(M) \rangle$ に対して、 $\theta(\alpha) \neq \theta^*(M)$ ならば、その系列における任意のリデックス出現uに対して、 $\exists v \in V_{\alpha_1}(\alpha, M), u \geq v$ である事を成立する。

(リダクションが α, M の共通部分のリデックスの集合に含まれていないである。)

【定義】：

リデックス・ペア $\langle \theta(\alpha), \theta^*(M) \rangle$ とは、 α は書き換え規則左辺であり、Mは書き換え規則の左辺或いはその部分項のペアである。

【証明】：

この証明は項のリダクションのリデックスが α, M の共通部分リデックスの集合に含まれている事を仮定して、矛盾を導く。

$\langle E, n+1 \rangle$ -オーバーラップの条件により：

$\alpha_1 \rightarrow \beta_1, \alpha_1 \rightarrow \beta_2 \in R$; 代入 θ, θ^* ; $\alpha_1/u = M \in V$

$\varepsilon \in O_{1, n}, T[\theta(\alpha_1) = \alpha \theta'(M)];$
 $\#T[\theta(\alpha_1) = \alpha \theta'(M)] \leq n+1$
 リダクションのリデックスが α , M の共通部分のリデックスの集合に含まれている仮定によって,

$\#T[\theta(\alpha_1)/v = \alpha \theta'(M)/v] = p \leq n$ であるような v を選ぶ事ができる. $v \in (O(\alpha_1) - O_v(\alpha_1)) \wedge (O(M) - O_v(M))$ かつ $v' \in (O(\alpha_1) - O_v(\alpha_1)) \wedge (O(M) - O_v(M)), v > v'$.

この時, R が $\langle E, n \rangle$ 非オーバーラップである条件によって, 規則 $\alpha_k \rightarrow \beta_k \in R$, 代入 σ, σ' ;

$T[\theta(\alpha_1)/v = \alpha \theta'(\alpha_1)/v] \geq T'$

$T' \equiv T[\sigma(\beta_k) \leftarrow \sigma(\alpha_k)] \cdot T[\sigma(\alpha_k) = \alpha \sigma'(\alpha_k)] \cdot$

$T[\sigma'(\alpha_k) \rightarrow \sigma'(\beta)]$

(ここで, $\sigma(\beta_k) \leftarrow \sigma(\alpha_k)$ は $\theta(\alpha) \rightarrow \theta'(M)$ のリダクション列の中の最も左のものである.)

$\#T[\sigma(\alpha_k) = \alpha \sigma'(\alpha_k)] \leq n$ と $\langle E, n \rangle$ 非オーバーラップ条件より, 削除補題(1)を適用して, $\sigma = \alpha \sigma'$ を導いた.

さらに, $\sigma(\beta_k) = \alpha \sigma'(\beta_k)$ を得た.

$T' \equiv T[\sigma(\beta_k) = \sigma'(\beta_k)]$ とすれば, $\#T' > \#T'$ であり, T' が T' で置き換える事と TRS R が右線形であることにより $\#T[\theta(\alpha_1)/v = \alpha \theta'(\alpha_1)/v] < p$ になり, これで矛盾を導いた. ■

[補題3]:

t ステップ後の集合 $A(t)$, $E_q(t)$, $B(t)$ に対して, 以下の事を成り立つ.

<1>. $\forall (x, P) \in B(t)$ に対して, $\theta(x) = \theta(P)$

<2>. $(x, P) \in B(t) \Rightarrow \forall (y, Q) \in B(t)$ に対して,
 $x \in V(Q)$.

(即ち, $B(t)$ のある要素の左辺に出現している変数は他の任意の要素の右辺に出現しない.)

<3>. $\forall (M, N) \in A(t)$, $E_q(t)$ に対して, $M \rightarrow N$ である.

<4>. $t+1$ ステップにおいて, このアルゴリズムが failuro を返す事がない.

[証明]:

この証明はアルゴリズム ω - $u n i f y$ の実行ステップに関する帰納法を用いる.

①. $t < 3$ の時, $B = \phi$ である, <1>, <2> が成立する.

$t = 1$ の時, 補題2によって, <3>, <4> も成立する.

②. $\forall (x, P) \in B(t)$, $A(t)$, $E_q(t)$ に対して,
 <1>, <2>, <3>, <4> が成立するである事を仮定する.

$t+1$ の時を討論する.

以下は(III)(IV)(V)(VI)(VII)(VIII)の順番に証明する.

(III)の場合: $(\exists (x, P), (x, Q) \in B(t))$

$B = \{B(t) - \{(x, P), (x, Q)\}\} \cup \{(x, P[u_1 \leftarrow N_1 \mid P/u_1 \equiv M_1, u_1 \in \text{Node}(P) \wedge \text{Node}(Q)]])\}$

$B(t+1) = \text{Insert}(B, \text{Front}(P, Q))$

$\text{Front}(P, Q) = \{(P/u, Q/u) \mid u \in O(P) \cap O(Q) \wedge ((P/u \in V) \vee (Q/u \in V))\}$

$(x, P) \in B(t)$, $(x, Q) \in B(t)$ なので, 帰納の仮定を用いて,

$\theta(x) = \theta(P)$ と $\theta(x) = \theta(Q)$

及び $x \in V(P)$ と $x \in V(Q)$ であるから:

$\theta(P) = \theta(Q)$, だから, $\theta(P/u) = \theta(Q/u)$

つまり: $\theta(x)/u_1 = \theta(x_1) = \theta(N_1)$

従って:

$\theta(x) = \theta(P[u_1 \leftarrow N_1 \mid \dots])$ つまり $\theta(x) = \theta(P)$

$x \in V(P(t+1))$ と $x \in V(Q(t+1))$ である事が分かった.

この(III)で A と E_q についての操作がなかった.

(IV)の場合: $(\exists (x, P) \in B(t), (x, Q) \in E_q(t))$

B に関する操作がないから, $B(t+1)$ が成立する.

帰納の仮定によって, $\text{Check}(P, Q) = \text{TURE}$ と $x \rightarrow Q$ である.

$E_q = (E_q - \{(x, Q)\}) \cup S(P, Q)$

$A = A \cup C(P, Q)$

補題2より, x と Q の間のリダクションは $\text{Front}(P, Q)$ により,

保たれてる. 従って, $\forall (M, N) \in \text{Front}(P, Q)$, $M \rightarrow N$

(V)の場合: $(\exists (x, P), (x, Q) \in E_q(t); P \rightarrow Q)$

$B(t+1) = \text{Insert}(B(t), \{(x, P[u_1 \leftarrow x_1 \mid P/u_1, Q/u_1 \equiv M_1 \vee N_1, u_1 \in V_{M_1 \vee N_1}(P, Q); \text{New}(x_1), 1 \leq i \leq |V_{M_1 \vee N_1}(P, Q)|]\})\})$

$\therefore \theta(P[u_1 \leftarrow x_1 \mid \dots]) = P[u_1 \leftarrow \theta(x_1) \mid \dots]$

$\theta(Q[u_1 \leftarrow x_1 \mid \dots]) = Q[u_1 \leftarrow \theta(x_1) \mid \dots]$

従って, $\theta(x) = \theta(P[u_1 \leftarrow \theta(x_1) \mid \dots])$ を示せばよい. その為に

$\theta(x)$ と $\theta(P)$, $\theta(Q)$ の間のリダクション出現 $u_1 \in V_{M_1 \vee N_1}(\theta(x)$,

$\theta(P)) \wedge V_{M_1 \vee N_1}(\theta(x), \theta(Q))$ を証明して, つまり: リダクション

が $\theta(x)$ と $\theta(Q)$ の共通部分のリデックスの集合に含まれていない事である.

これを証明するために TRS が制限的である条件を必要とすることを利用する.

帰納法で証明するために, リダクションがフロンティア出現以外で行う事を仮定する. 即ち:

$v \in O(\theta(P))$, $u \in V_{M_1 \vee N_1}(\theta(x), \theta(P))$, s. t. $v < u$.

規則 $\alpha_k \rightarrow \beta_k \in R$

制限条件によって, x の個数が二個(right liner)であるから:

$\theta'(x) = \theta(x)[v \leftarrow \sigma(\beta_k)]$

$\theta'(x) = \theta(x)[v \leftarrow \sigma'(\beta_k)]$

項書き換えシステム R が $\langle E, n+1 \rangle$ オーバーラップであるから,

入力項 $\theta(\alpha_1) = \alpha \theta(M)$ のリダクション・ステップ数:

$\#T[\theta(\alpha_1) = \alpha \theta(M)] \leq n+1$

従って:

$\theta'(\alpha_1) = \alpha_1[w \leftarrow \theta'(x) \mid \alpha_1/w = x]$

$\theta'(M) = M[w' \leftarrow \theta'(x) \mid M/w' = x]$ によって,

$\#T[\theta'(\alpha_1) = \alpha \theta'(M)] \leq n$ になり,

これは矛盾であるから, リダクションがそれぞれのフロンティア出現以上で行う事が明かとなる:

$\theta(x) = \theta(P)$ と $\theta(x) = \theta(Q)$ である.

そして, P のフロンティアの変数を全部新しい変数に換えられたから, $x \in V(P)$ である事も明かとなる.

帰納の仮定によって, $x \rightarrow P$, $x \rightarrow Q$ だから, 補題2によって,

$\text{Check}(P, Q) = \text{TURE}$ である. そして, (IV)で示したように

$(M, N) \in \text{Front}(P, Q)$, $M \rightarrow N$

(VI)の場合: $(\exists x \rightarrow y, (x, P) \in B(t), (y, Q) \in E_q(t); x \neq y)$

$V(P) \cup V(Q) = \phi$ ならば,

$\theta(x) = \theta(P) = \theta(Q) = \theta(y)$ である.

$V(P) \cup V(Q) \neq \phi$ の時,

$B(t+1) = \text{Insert}(B(t), \{(y, Q[u_1 \leftarrow y_1 \mid P/u_1, Q/u_1 \equiv M_1 \vee N_1,$

$u_j \in V_{n+1}(P, Q); \text{New}(y_j), 1 \leq j \leq |V_{n+1}(P, Q)| \})$
 リダクション系列: $\theta(Q) \rightarrow \theta(y) \rightarrow \theta(x) = \theta(P)$ において、
 制限条件により、変数 y の数が高々の二個出あるから、 $\theta(y)$
 からのリダクションがあれば、一つ $\theta(Q)$ へ向いて、もう一方
 $\theta(x)$ つまり $\theta(P)$ に向く。従って、これと (V) 番は同じで
 あるから、そのような方法で、
 $\theta(y)/u_j = \theta(y_j) = \theta(x_j) = \theta(x)/u_j$ が明かとなる。
 従って、 $\theta(y) = \theta(Q)$ と $y \in V(Q)$ である。
 帰納の仮定によって、 $x = P, y \rightarrow Q, x \rightarrow y$ だから、補題 2 によって、
 $\text{Check}(P, Q) = \text{TRUE}$ である。そして、(IV) で示したように
 $(M, N) \in \text{Front}(P, Q), M \rightarrow N$
 (VII) の場合: $(\exists x \rightarrow y, (x, P) \in B(t), (y, Q) \in B(t), P \neq Q)$
 $B(t+1) := \text{Insert}(B(t), \{(z, P[u_j \leftarrow z_j | P/u_j, Q/u_j \equiv M_j \vee N_j,$
 $u_j \in V_{n+1}(P, Q); \text{New}(z_j), 1 \leq j \leq |V_{n+1}(P, Q)| \})$)

帰納の仮定によって、 t の時:

$\theta(x) = \theta(P), \theta(y) = \theta(Q)$ である。

つまり: $\theta(P) = \theta(x) \rightarrow \theta(z) \rightarrow \theta(y) = \theta(Q)$

制限条件により、変数 z の数が高々の二個出あるから、 $\theta(z)$
 からのリダクションがあれば、一つ $\theta(y)$ つまり $\theta(Q)$ へ向いて、
 もう一方 $\theta(x)$ つまり $\theta(P)$ に向く。従って:

以下のような v が存在する事を仮定する。

$v \in (O(\theta(P)) - V_{n+1}(\theta(x), \theta(P))) \cap (O(\theta(Q)) - V_{n+1}(\theta(y),$
 $\theta(Q)))$; 代入 σ, σ' ; 規則 $\alpha \rightarrow \beta$

$\theta(x) = \theta(z) [v \leftarrow \sigma(\beta)]$

$\theta'(x) = \theta(z) [v \leftarrow \sigma'(\beta)]$

TRS R が ω -非 ω - γ であるから、

入力項 $\theta(\alpha_j) = \theta(M)$ の $\#T[\theta(\alpha_j) = \theta(M)] \leq n+1$

$\theta'(\alpha_j) = \alpha_j [w \leftarrow \theta(z) | \alpha_j/w = z]$

$\theta'(M) = M[w' \leftarrow \theta'(z) | M/w' = z]$ によって、

$\#T[\theta'(\alpha_j) = \theta'(M)] \leq n$ になり、

これは矛盾であるから、リダクションがそれぞれのフロンティア
 出現以上で行う事が明かとなる:

$\theta(z) = \theta(P)$ である。

そして、 P のフロンティアの変数を全部新しい変数に換えられた
 から、 $z \in V(P)$ である事も明かとなる。

帰納の仮定によって、 $x = P, y = Q, x \rightarrow z \rightarrow y$ だから、補題 2 によっ

て、 $\text{Check}(P, Q) = \text{TRUE}$ である。そして、(IV) で示したように

$(M, N) \in \text{Front}(P, Q), M \rightarrow N$

(VIII) の場合: $(\exists x \rightarrow y, (x, P) \in \text{Eq}(t), (y, Q) \in \text{Eq}(t), x \neq y)$

$V(P) \cup V(Q) = \emptyset$ の時、 $P \equiv Q$ ならば、

$\theta(x) \rightarrow \theta(P)$ と $\theta(y) \rightarrow \theta(Q)$ より、

$\theta(x) = \theta(P) = P$ と $\theta(y) = \theta(Q) = Q$

が得られた。

$V(P) \cup V(Q) \neq \emptyset$ の時、

$B(t+1) := \text{Insert}(B(t), \{(x, P[u_j \leftarrow a_j | P/u_j, Q/u_j \equiv M_j \vee N_j,$
 $u_j \in V_{n+1}(P, Q); \text{New}(a_j), 1 \leq j \leq |V_{n+1}(P, Q)| \}),$

$(y, Q[u_j \leftarrow b_j | P/u_j, Q/u_j \equiv M_j \vee N_j, u_j \in V_{n+1}$

$(P, Q); \text{New}(b_j), 1 \leq j \leq |V_{n+1}(P, Q)| \})$)

帰納の仮定によって、 t の時:

$\theta(x) \rightarrow \theta(P), \theta(y) \rightarrow \theta(Q)$ である。

つまり: $\theta(P) \rightarrow \theta(x) \rightarrow \theta(y) \rightarrow \theta(Q)$

制限条件により変数 x, y の数が高々の二個出あるから、 $\theta(y)$
 からのリダクションがあれば、一つ $\theta(Q)$ へ向いて、もう一方
 $\theta(x)$ つまり $\theta(P)$ に向く。 $\theta(x)$ からのリダクションがあれば、
 一つ $\theta(P)$ へ向いて、もう一方 $\theta(y)$ つまり $\theta(Q)$ に向く。
 従って、これと (V) 番は同じであるから、そのような方法を
 二回適用すると、以下の事になる:

$\theta(a_j) = \theta(x/u_j)$ と $\theta(b_j) = \theta(y/u_j)$ により、

$\theta(x) = \theta(P)$ と $\theta(y) = \theta(Q)$ と $x \in V(P)$ と $y \in V(Q)$ であ
 る事が分かった。

帰納の仮定によって、 $x \rightarrow P, y \rightarrow Q, x \rightarrow y$ だから、補題 2 によっ

て、 $\text{Check}(P, Q) = \text{TRUE}$ である。そして、(IV) で示したように

$(M, N) \in \text{Front}(P, Q), M \rightarrow N$

これで、この補題の証明を終了する。■

5. 唯一正規性と合流性

TRS が ω -非 ω - γ であるかどうか判定する問題は可解で
 あるから、定理 1 より、 E -非 ω - γ 性を保証する判定可能な
 一つの十分条件を与えることが出来た。この十分条件は文献[3]
 の結果より、唯一正規性を保証するための十分条件でもある。

また、本研究で与えた定理より、非線形かつ非停止の TRS の
 合流性を保証する判定可能な十分条件を与えることが出来る。即
 ち、合流性に関する次の結果が知られている。

[定理 2] ([5], [6])

TRS が E -非 ω - γ かつ右定数項システムであるならば、合
 流性を持つ。

従って、この定理 2 と本論文の定理 1 より、次の系が得られる。

[系] TRS が ω -非 ω - γ システムかつ制限的な右定数項シ
 ステムであるならば、合流性を持つ。

参考文献

- [1] 坂井公: Knuth-Bendix の完備化手続きとその
 応用, コンピュータソフトウェア, (pp 2-22) 1987, 1.
- [2] 二本厚吉, 外山芳人: 項書き換え型計算モデルとその応用,
 情報処理, Vol. 24, No. 2, pp. 133-146, 1983.
- [3] Mizuhito OGAWA and Satoshi ONO: On the Uniquely Con
 verging Property of Nonlinear Term Rewriting System,
 ソフトウェア基礎論 1989, 5, (pp 61-70)
- [4] Huet, G.: Confluent reductions: abstract propertie
 s and applications to term rewriting systems,
 J. ACM 27-4, pp. 797-821 (Oct. 1980).
- [5] Oyamauchi, M.: Some results on decision problems
 for right-ground term-rewriting systems, Proc. of
 Toyohashi Symposium on Theoretical Computer Science
 1990, (pp 41-42)
- [6] Oyamauchi, M.: On the confluence problem for right
 -ground term-rewriting systems, unpublished manuscript.
- [7] 張 磊: 項書き換えシステムに関する研究, 三重大学
 修士論文, 1990.
- [8] 張磊 大田通夫 伊藤英明: 非線形項書き換えシステム
 の唯一正規形の性質, 第 43 回情報処理学会全国大会講演論文集
 (3), pp. 75-76.